**CSE537 Artificial Intelligence**

**Project 2 – Game Search (Connect Four)**

**Team Details**

**Name SBU ID Net ID**

Aman Raj 109939872 amraj

Dhanendra Jain 110369635 dhjain

For running - Run lab3.py

**Q1. basic\_player vs human\_player**

Uncomment line no. 45 in lab3.py –

run\_game(basic\_player, human\_player)

**basic\_player wins**

Player 1 (X) puts a token in column 2

Win for X!

0 1 2 3 4 5 6

0 X

1 X X

2 O X X

3 X X O

4 O O X X

5 O O O X O O

Running time: 40.9843786351

Expanded nodes in new player: 21403

**Q2. new\_player vs basic\_player**

Uncomment line no. 53 in lab3.py –

run\_game(new\_player, basic\_player, winNum=4)

**new\_player wins**

Win for X!

Player 1 (X) puts a token in column 0

0 1 2 3 4 5 6

0 X X O

1 O X X

2 X X X

3 X O O

4 X O O X O

5 O O O X X O

Running time: 20.9145311025

Expanded nodes in new player: 39216

**Q3. alphabeta\_player vs basic\_player**

Uncomment line no. 187 in lab3.py –

run\_game(alphabeta\_player, basic\_player, winNum=4)

**alphabeta\_player wins**

Player 1 (X) puts a token in column 3

Win for X!

0 1 2 3 4 5 6

0 X

1 X

2 X O

3 O O O

4 O O O

5 X O X X X X X

Running time: 9.90694622159

Expanded nodes in apha-beta: 3735

**Generalization**

**Connect-k** is implemented. We passed an extra k\_value argument in ConnectFourBoard Class.

Used new variable in constructor – “self.win\_num = win\_num”

Default value of win\_num is 4.

For changing its value – call run\_game() with a desired value of k.

Eg. - For k =3

**For new\_player vs basic\_player,** uncomment line no. 54 in lab3.py –

run\_game(new\_player, basic\_player, winNum=3)

**For alphabeta\_player vs basic\_player,** uncomment line no. 188 in lab3.py –

run\_game(alphabeta\_player, basic\_player, winNum=3)

alphabeta\_player wins

Win for X!

0 1 2 3 4 5 6

0 X

1 X

2 X

3 O

4 O O

5 X O X

Running time: 5.09540502306

Expanded nodes in apha-beta: 2041

**New Evaluate Function Description**

We are calculating score for top cell element in every column.

First, get the index of the first cell in the specified column that is filled using function - get\_height\_of\_column()

For every top cell in every column, we calculate the maximum length using that cell (in all directions – horizontal, vertical, left diagonal, right diagonal) using function –

\_max\_length\_from\_cell()

Assigned score of 100 if length is 3

25 if length is 2

5 if length is 1

We calculate scores for current player and opponent player separately. And reduce the total score of opponent player from the final score.

**Note –** For alpha-beta search taken reference from

<https://en.wikipedia.org/wiki/Alpha-beta_pruning>